BLM2041 Signals and Systems

Week 4

The Instructors:

Prof. Dr. Nizamettin Aydın

naydin@yildiz.edu.tr

Asist. Prof. Dr. Ferkan Yilmaz

ferkan@yildiz.edu.tr

Systems Properties

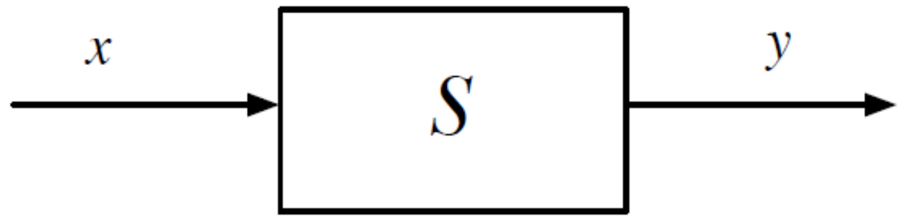
Systems

- A system transforms *input signals* into *output signals*.
- A system is a *function* mapping input signals into output signals.
- We will concentrate on systems with one input and one output *i.e.* *single-input, single-output* (SISO) systems.
- Notation:
  - $y = Sx$ or $y = S(x)$, meaning the system S acts on an input signal x to produce output signal y.
  - $y = Sx$ does not (in general) mean multiplication!

Systems Properties

Block diagrams

Systems often denoted by *block diagram*:



- Lines with arrows denote signals (*not* wires).
- Boxes denote systems; arrows show inputs & outputs.
- Special symbols for some systems.

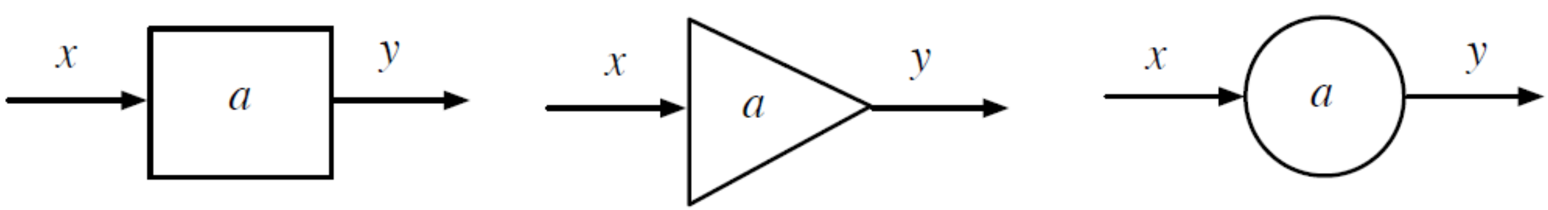
Systems Properties

Examples

(with input signal x and output signal y)

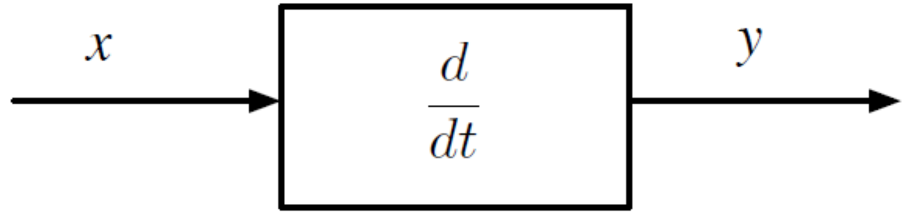
Scaling system: $y(t) = ax(t)$

- Called an *amplifier* if $|a| > 1$.
- Called an *attenuator* if $|a| < 1$.
- Called *inverting* if $a < 0$.
- a is called the *gain* or *scale factor*.
- Sometimes denoted by triangle or circle in block diagram:



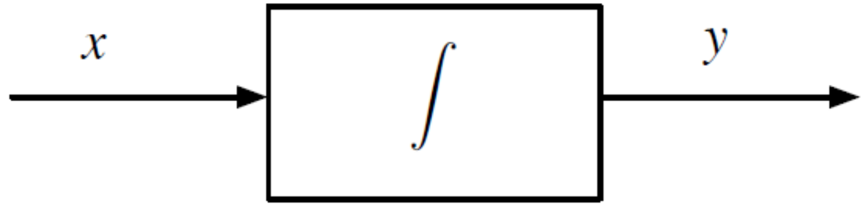
Systems Properties

Differentiator: $y(t) = x'(t)$



Integrator: $y(t) = \int_a^t x(\tau)\, d\tau$ (a is often 0 or $-\infty$)

Common notation for integrator:



Systems Properties

time shift system: $y(t) = x(t - T)$

- called a *delay system* if $T > 0$
- called a *predictor system* if $T < 0$

Systems Properties

convolution system:

$$y(t) = \int x(t - \tau) h(\tau)\, d\tau,$$

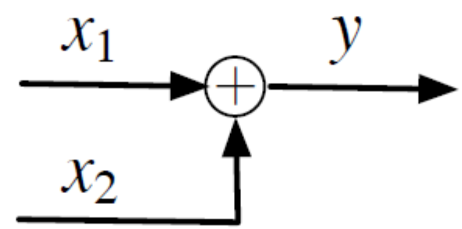
where h is a given function (you'll be hearing much more about this!)

Systems Properties

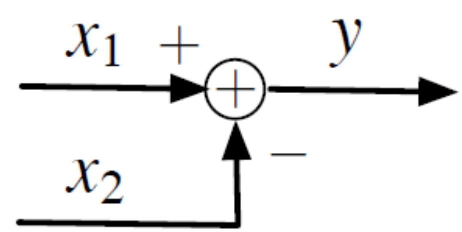
Examples with multiple inputs

Inputs $x_1(t)$, $x_2(t)$, and Output $y(t)$)

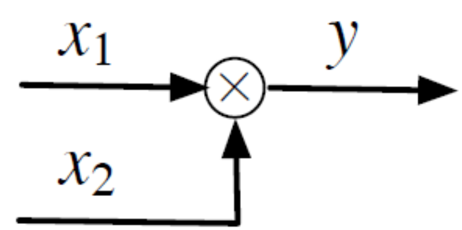
- **summing system:** $y(t) = x_1(t) + x_2(t)$



- **difference system:** $y(t) = x_1(t) - x_2(t)$



- **multiplier system:** $y(t) = x_1(t)x_2(t)$

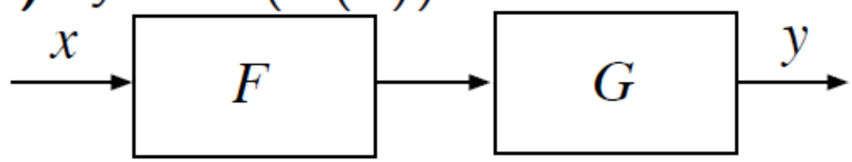


Systems Properties

Interconnection of Systems

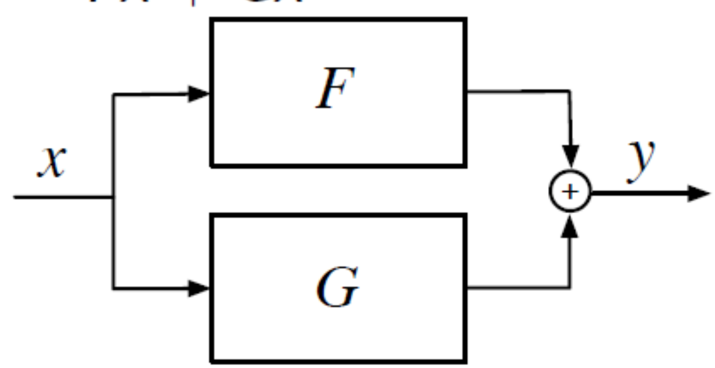
We can interconnect systems to form new systems,

- **cascade (or series):** $y = G(F(x)) = GFx$



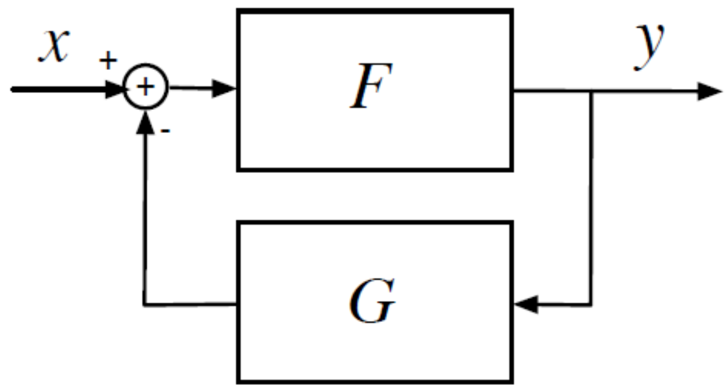
(note that block diagrams and algebra are *reversed*)

- **sum (or parallel):** $y = Fx + Gx$



Systems Properties

- **feedback:** $y = F(x - Gy)$

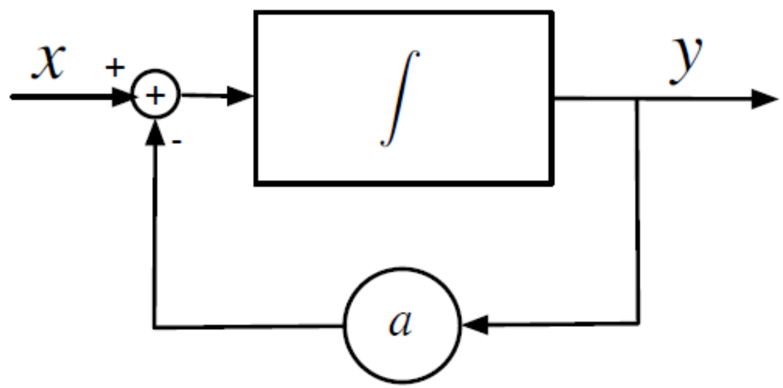


In general,

- Block diagrams are a symbolic way to describe a connection of systems.
- We can just as well write out the equations relating the signals.
- We can go back and forth between the system block diagram and the system equations.

Systems Properties

Example: Integrator with feedback



Input to integrator is $x - ay$, so

$$\int^{t} (x(\tau) - ay(\tau))\, d\tau = y(t)$$

Another useful method: the *input* to an integrator is the derivative of its output, so we have

$$x - ay = y'$$

Systems Properties

Linearity

A system F is **linear** if the following two properties hold:

1. **homogeneity:** if x is any signal and a is any scalar,

$$F(ax) = aF(x)$$

2. **superposition:** if x and $\tilde{x}$ are any two signals,

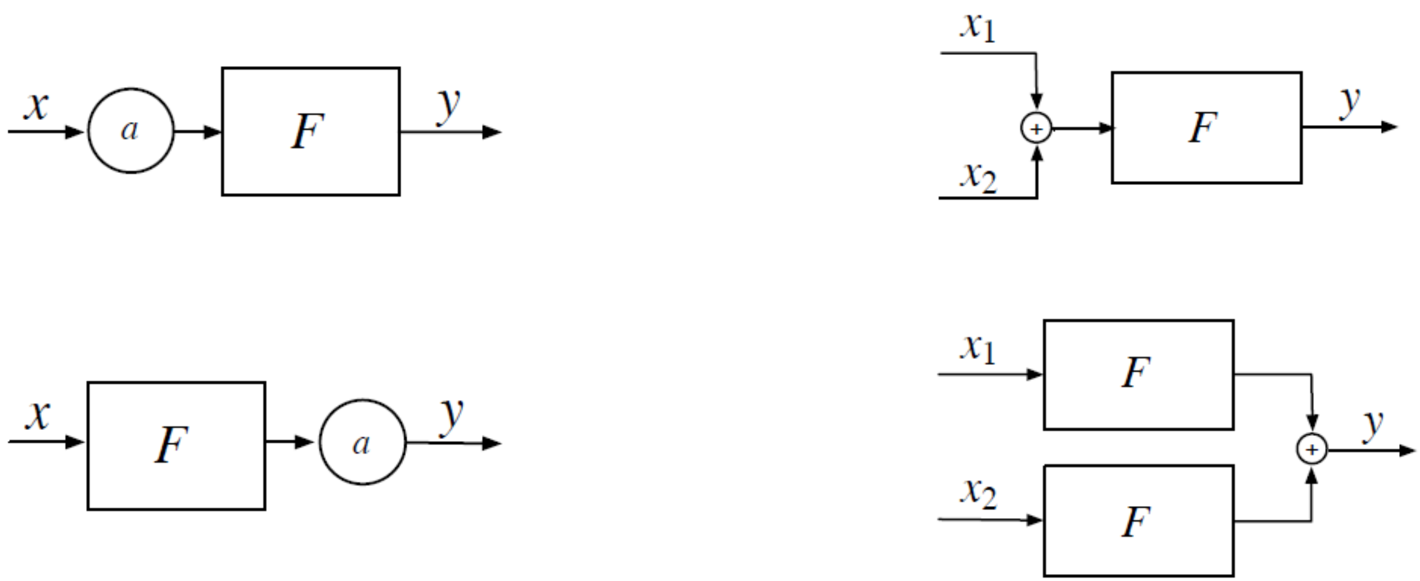
$$F(x + \tilde{x}) = F(x) + F(\tilde{x})$$

In words, linearity means:

- Scaling before or after the system is the same.
- Summing before or after the system is the same.

Systems Properties

Linearity means the following pairs of block diagrams are equivalent, *i.e.*, have the same output for any input(s)

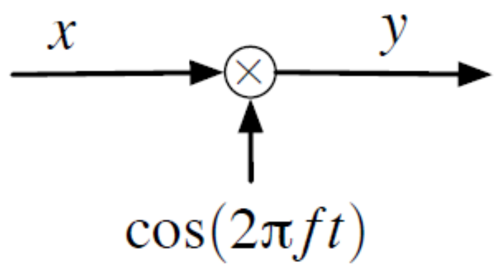


Examples of linear systems: scaling system, differentiator, integrator, running average, time shift, convolution, modulator, sampler.

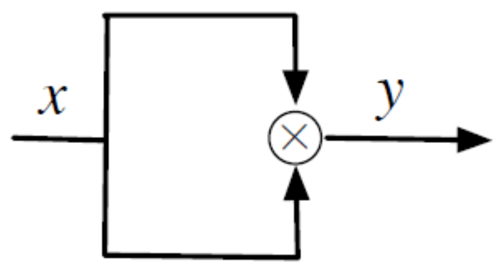
Examples of nonlinear systems: sign detector, multiplier (sometimes), comparator, quantizer, adaptive filter

Systems Properties

- Multiplier as a modulator, $y(t) = x(t)\cos(2\pi ft)$, is *linear*.



- Multiplier as a squaring system, $y(t) = x^2(t)$ is *nonlinear*.



Systems Properties

System Memory

- A system is *memoryless* if the output depends only on the present input.
  - Ideal amplifier
  - Ideal gear, transmission, or lever in a mechanical system
- A *system with memory* has an output signal that depends on inputs in the past or future.
  - Energy storage circuit elements such as capacitors and inductors
  - Springs or moving masses in mechanical systems
- A *causal* system has an output that depends only on past or present inputs.
  - Any real physical circuit, or mechanical system.

Systems Properties

Time-Invariance

- A system is time-invariant if a time shift in the input produces the same time shift in the output.
- For a system F,

  $$y(t) = Fx(t)$$

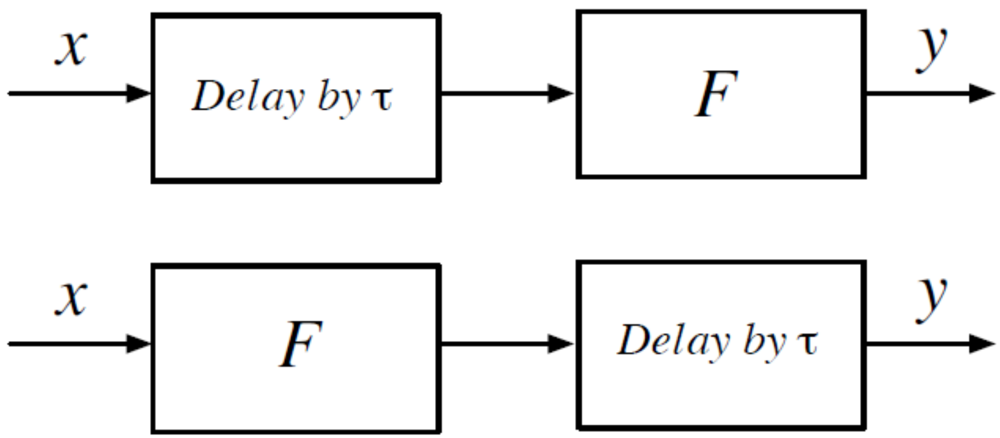
  implies that

  $$y(t-\tau) = Fx(t-\tau)$$

  for any time shift τ.

Systems Properties

- Implies that delay and the system F commute. These block diagrams are equivalent:



- Time invariance is an important system property. It greatly simplifies the analysis of systems.

Systems Properties

System Stability

- Stability important for most engineering applications.
- Many definitions
- If a bounded input

$$|x(t)| \leq M_x < \infty$$

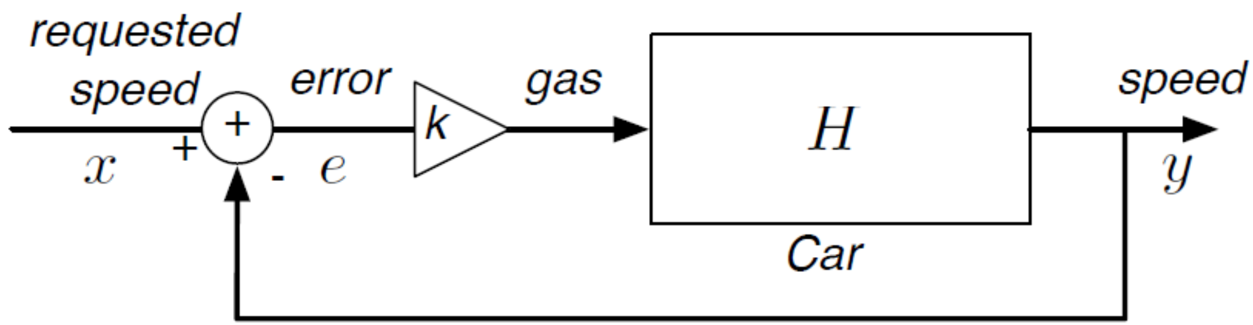
always results in a bounded output

$$|y(t)| \leq M_y < \infty,$$

where M_x and M_y are finite positive numbers, the system is *Bounded Input Bounded Output (BIBO) stable*.

Systems Properties

Example: Cruise control, from introduction,



The output y is

$$y = H(k(x - y))$$

We'll see later that this system can become unstable if k is too large (depending on H)

- Positive error adds gas
- Delay car velocity change, speed overshoots
- Negative error cuts gas off
- Delay in velocity change, speed undershoots
- Repeat!

Systems Properties

System Invertibility

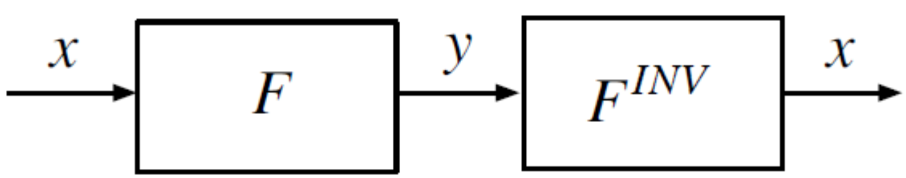
- A system is invertible if the input signal can be recovered from the output signal.
- If F is an invertible system, and

$$y = Fx$$

then there is an inverse system F^{INV} such that

$$x = F^{INV} y = F^{INV} F x$$

so $F^{INV} F = I$, the identity operator.



Systems Properties

Systems Described by Differential Equations

Many systems are described by a *linear constant coefficient ordinary differential equation* (LCCODE):

$$a_n y^{(n)}(t) + \cdots + a_1 y'(t) + a_0 y(t) = b_m x^{(m)}(t) + \cdots + b_1 x'(t) + b_0 x(t)$$

with given *initial conditions*

$$y^{(n-1)}(0), \quad \ldots \quad , y'(0), \quad y(0)$$

(which fixes $y(t)$, given $x(t)$)

- n is called the *order* of the system
- $b_0, \ldots, b_m, a_0, \ldots, a_n$ are the *coefficients* of the system

Systems Properties

This is important because LCCODE systems are **linear** when initial conditions are all zero.

- Many systems can be described this way
- If we can describe a system this way, we know it is linear

Note that an LCCODE gives an *implicit* description of a system.

- It describes how $x(t)$, $y(t)$, and their derivatives interrelate
- It doesn't give you an explicit solution for $y(t)$ in terms of $x(t)$

Soon we'll be able to *explicitly* express $y(t)$ in terms of $x(t)$

Systems Properties

Examples

Simple examples

- scaling system ($a_0 = 1,\ b_0 = a$)

$$y = ax$$

- integrator ($a_1 = 1,\ b_0 = 1$)

$$y' = x$$

- differentiator ($a_0 = 1,\ b_1 = 1$)

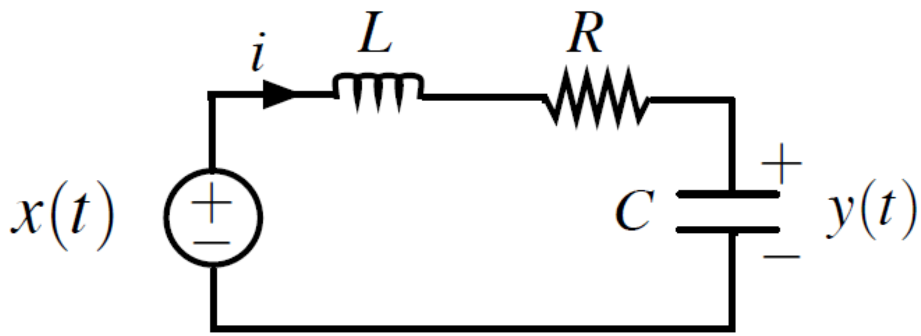
$$y = x'$$

- integrator with feedback (a few slides back, $a_1 = 1, a_0 = a, b_0 = 1$)

$$y' + ay = x$$

Systems Properties

2nd Order Circuit Example



By Kirchoff's voltage law

$$x - Li' - Ri - y = 0$$

Using $i = Cy'$,

$$x - LCy'' - RCy' - y = 0$$

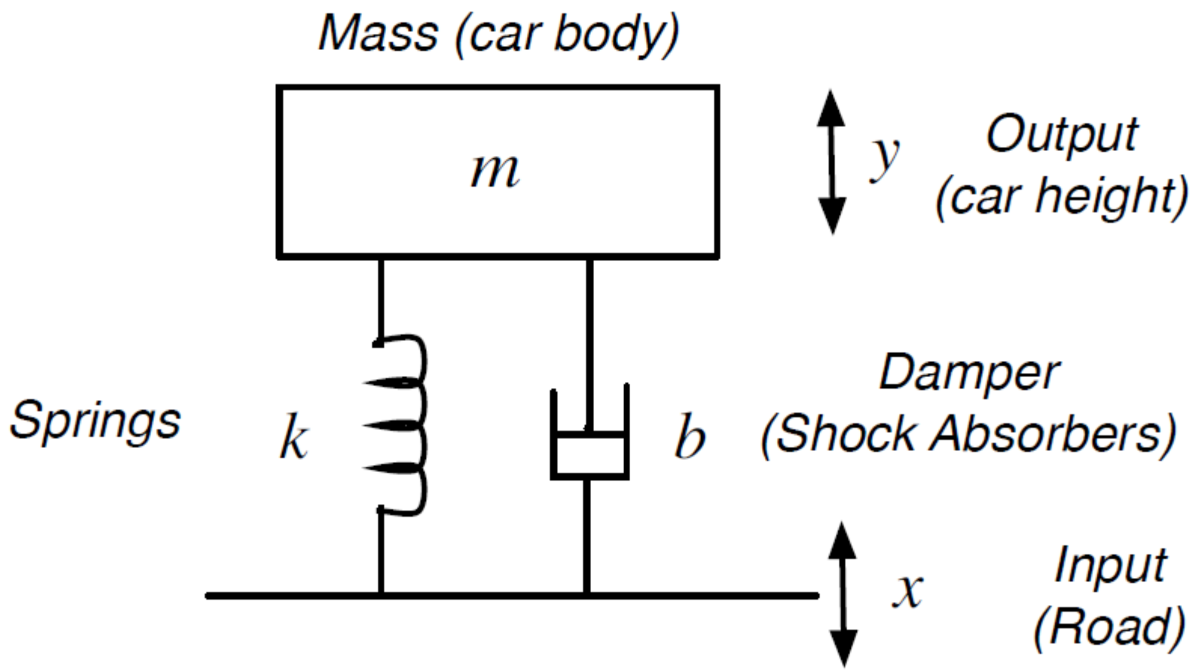
or

$$LCy'' + RCy' + y = x$$

which is an LCCODE. This is a linear system.

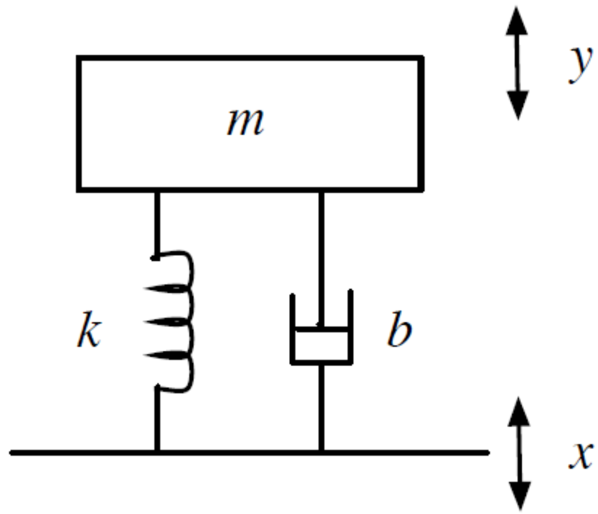
Systems Properties

Mechanical System



This can represent suspension system, or building during earthquake, ...

Systems Properties



- $x(t)$ is displacement of base; $y(t)$ is displacement of mass
- spring force is $k(x-y)$; damping force is $b(x-y)'$
- Newton's equation is $my'' = b(x-y)' + k(x-y)$

Rewrite as second-order LCCODE

$$my'' + by' + ky = bx' + kx$$

Systems Properties

Discrete-Time Systems

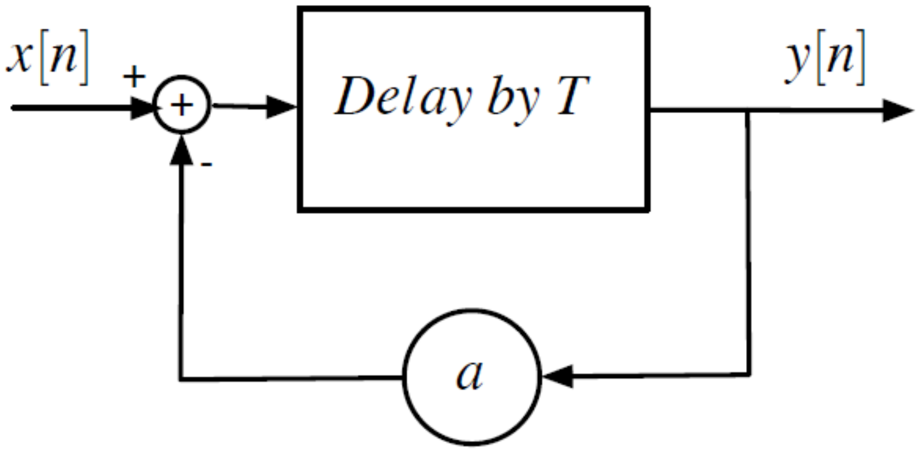
- Many of the same block diagram elements
- Scaling and delay blocks common
- The system equations are *difference equations*

$$a_0 y[n] + a_1 y[n-1] + \ldots = b_0 x[n] + b_1 x[n-1] + \ldots$$

where $x[n]$ is the input, and $y[n]$ is the output.

Systems Properties

Discrete-Time System Example



- The input into the delay is

$$e[n] = x[n] - ay[n]$$

- The output is $y[n] = e[n-1]$, so

$$y[n] = x[n-1] - ay[n-1].$$